WL-TR-92-1123

A MARKET-ORIENTED PROGRAMMING ENVIRONMENT AND ITS APPLICATION TO DISTRIBUTED MULTICOMMODITY FLOW PROBLEMS

Michael P. Wellman

AVIONICS DIRECTORATE
ARTIFICIAL INTELLIGENCE TECHNOLOGY OFFICE
WRIGHT-PATTERSON AFB OH 45433-6543

OCTOBER 1992

FINAL REPORT FOR PERIOD AUGUST 1989 - JULY 1992

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB OH 45433-6543

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

WILLIAM R. BAKER
Chief, Artificial Intelligence
Technology Office

STEPHEN G. PETERS, Major, USAF
Acting Chief, Information Processing
Technology Branch

Charles H Krueger
CHARLES H. KRUEGER
Chief, System Avionics Divison

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/AAA WPAFB, OH 45433-6543 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
OCT 1992

3. REPORT TYPE AND DATES COVERED
FINAL AUGUST 1989 - JULY 1992

4. TITLE AND SUBTITLE
A MARKET-ORIENTED PROGRAMMING
ENVIRONMENT AND ITS APPLICATION TO
DISTRIBUTED MULTICOMMODITY FLOW PROBLEMS

5. FUNDING NUMBERS

C	
PE	62204
PR	TXI0
TA	00
WU	03

6. AUTHOR(S)
Michael P. Wellman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
AVIONICS DIRECTORATE
ASC ARTIFICIAL INTELLIGENCE TECH OFFICE
WRIGHT-PATTERSON AFB OH 45433-6543

8. PERFORMING ORGANIZATION REPORT NUMBER
WL-TR-92-1123

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AVIONICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT PATTERSON AFB OH 45433-6543
WL/AAA, Attn: BAKER 513-2555800

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-92-1123

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Market price mechanisms from economics provide a common and well-studied framework for coordinating decentralized decision processes with minimal communication. WALRAS is a general "market-oriented programming" environment for the construction and analysis of distributed resource allocation systems, based on general-equilibrium theory. The environment provides basic constructs for defining computational market structures, and a procedure for deriving their corresponding competitive equilibria. In a particular realization of this approach for a form of multicommodity flow problem, we see that careful construction of the decision process according to economic principles can lead to effective decentralization, and that the behavior of the system can be meaningfully analyzed in economic terms.

14. SUBJECT TERMS

15. NUMBER OF PAGES
31

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

1 The WALRAS Project

This work arose from the observation that real-world, large-scale operations planning—as in military transportation scheduling—is highly decentralized, and the conviction that automated decision-support systems serving such tasks should reflect the distributed nature of the problem. Tackling such problems from the "headquarters perspective" can often be misleading, and at worst, the failure to reflect realities in informational and authority structure can produce results that are useless in managing the actual operation as it unfolds.

Taking the decentralized view seriously, the focus on coordination mechanisms follows naturally. Market price schemes are one obvious alternative, which bring the notable advantages of (1) demonstrated ability to achieve useful results in real-world decentralized decision making, (2) extensive theoretical development and analytical machinery, and (3) known limitations. Moreover, the use of economic transactions provides a link between internal coordination of computational modules and coordination with agents external to the computer (i.e., real-world organizations). Market mechanisms have been studied somewhat in the computational realm, but investigations of this sort have been limited to date.

The WALRAS project was initiated in July 1991 in order to systematically study the potential of market mechanisms in a computational environment for decentralized decision-making. The project was conceived as a domain-independent framework, but initial explorations were focused for concreteness on transportation planning, the problem initially motivating the work.

The resulting system is a prototype environment for what I call "market-oriented programming." It has been tested on some simplified transportation planning, and ongoing work (by the author and Jon Doyle) is exploring both more elaborate models and totally different problem domains. The initial results, described in the following report, indicate a system that can support development of decentralized resource allocation systems, which produce optimal results under certain idealized conditions. Although this line of research is still in fairly early stages, the experience to date suggests that market-based approaches provide a promising direction for development of decentralized decision-support systems in the future.

2 Distributed Resource Allocation

In a *distributed* or multiagent planning system, the plan for the system as a whole is a composite of plans produced by its constituent agents. Planning might be distributed because agents are separated geographically, have different information, possess distinct capabilities or authority, or were designed and implemented separately. In any case, because each agent has limited competence and awareness of the decisions produced by others, some sort of coordination is required to maximize the performance of the overall system. However, central control or extensive communication is deemed infeasible, as it violates whatever constraints dictated distribution of the planning task in the first place.

The task facing the designer of a distributed planning system is to define a computationally efficient coordination mechanism and its realization for a given configuration of agents (in some cases the configuration itself is a design parameter). By the term *agent*, I refer to a module that acts within the mechanism according to its own knowledge and interests. The capabilities of the agents and their organization in an overall decision-making structure determine the behavior of the system as a whole. Because it concerns the collective behavior of self-interested decision makers, the design of this decentralized structure is fundamentally an exercise in economics. The problem of developing architectures for distributed planning is largely one of *mechanism design* [1, 2], and many ideas and results from economics are directly applicable. In particular, the class of mechanisms based on price systems and competition has been deeply investigated by economists, who have characterized the conditions for its efficiency and compatibility with other features of the economy. When applicable, the competitive mechanism achieves coordination with minimal communication requirements (in a precise sense related to the dimensionality of messages transmitted among agents [2]).

The theory of *general equilibrium* [3] provides the foundation for a general approach to the construction of distributed planning systems based on price mechanisms. In this approach, we regard the constituent planning agents as consumers and producers in an artificial economy, and define their individual activities in terms of production and consumption of commodities. Interactions among agents are cast as exchanges, the terms of which are mediated by the underlying economic mechanism, or protocol. By specifying the universe of commodities, the configuration of agents, and the interaction protocol, we can achieve a variety of interesting anJ often effective decentralized behaviors. Furthermore, we can apply economic theory to the analysis of alternative architectures, and thus exploit a wealth of existing knowledge in the design of distributed planners.

In the following, I describe this general approach and a programming environment based on it. An example problem in distributed transportation planning demonstrates the feasibility of decentralizing a problem with nontrivial interactions, and the applicability of economic principles to collective problem solving.

3 Example: Multicommodity Flow

In a simple version of the multicommodity flow problem, the task is to allocate a given set of cargo movements over a given transportation network. The transportation network is a collection of locations, with links (directed edges) identifying feasible transportation operations. Associated with each link is a specification of the cost of moving cargo along it. Suppose further that the cargo is homogeneous, and amounts of cargo are arbitrarily divisible. A movement requirement associates an amount of cargo with an origin-destination pair. The planning problem is to determine the amount to transport on each link in order to move all the cargo at the minimum cost.[1]

A distributed version of the problem would decentralize the responsibility for transporting separate cargo elements. For example, planning modules corresponding to geographically or organizationally disparate units might arrange the transportation for cargo within their respective spheres of authority. Or decision-making activity might be decomposed along hierarchical levels of abstraction, gross functional characteristics, or according to any other relevant distinction. This decentralization might result from real distribution of authority within a human organization, from inherent informational asymmetries and communication barriers, or from modularity imposed to facilitate software engineering.

Consider, for example, the abstract transportation network of Figure 1, taken from Harker [4].[2] There are four locations, with directed links as shown. Consider two movement requirements. The first is to transport cargo from location 1 to location 4, and the second in the reverse direction. Suppose we wish to decentralize authority so that separate agents (called shippers) decide how to allocate the cargo for each movement. The first shipper decides how to split its cargo units between the paths 1 → 2 → 4 and 1 → 2 → 3 → 4, while

[1] This simplification ignores salient aspects of real transportation planning. For instance, this model is completely atemporal, and is hence more suitable for planning steady-state flows than for planning dynamic movements.

[2] Models of this sort are employed in transportation analysis to predict cargo movements and hence characterize the effect of variations in transportation infrastructure or policy. Their intent is descriptive, as the agents are private individuals or firms outside the policymaker's control. Although the overall role of a planning system is to *prescribe* behavior, the designer of a distributed architecture also requires a descriptive model of modules behavior to characterize the effect of alternative configurations and coordination mechanisms.

the second figures the split between paths 4 → 2 → 1 and 4 → 2 → 3 → 1. Note that the latter paths for each shipper share a common resource: the link 2 → 3.

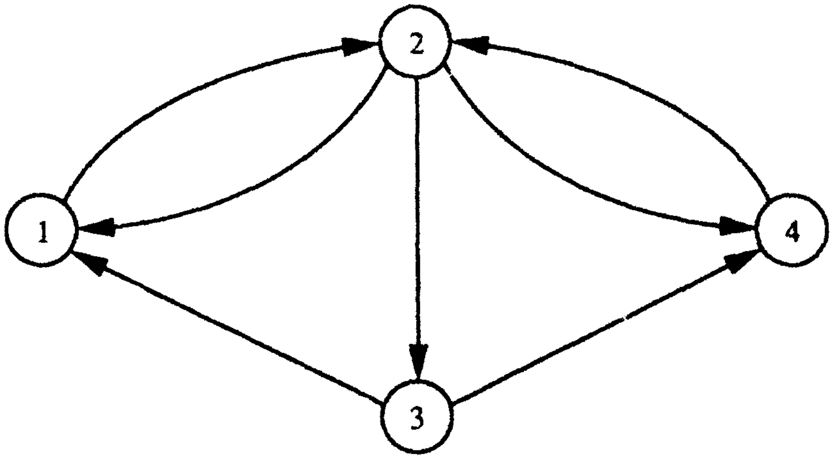

Figure 1: A simple network (from Harker [4]).

Because of their overlapping resource demands, the shippers' decisions appear to be necessarily intertwined. In a congested network, for example, the cost for transporting a unit of cargo over a link is increasing in the overall usage of the link. A shipper planning its cargo movements as if it were the only user on a network would thus underestimate its costs and potentially misallocate transportation resources.

For the analysis of networks such as this, transportation researchers have developed equilibrium concepts describing the collective behavior of the shippers. In a *system equilibrium*, the overall transportation of cargo proceeds as if there were an omniscient central planner directing the movement of each shipment so as to minimize the total aggregate cost of meeting the requirements. In a *user equilibrium*, the overall allocation of cargo movements minimizes each shipper's total cost, with shippers sharing proportionately the cost of shared resources. There are also some intermediate possibilities, corresponding to game-theoretic equilibrium concepts such as the Nash equilibrium, where each shipper behaves optimally given the transportation policies of the remaining shippers [5].[3]

From our perspective as designer of the distributed planner, we seek a decentralization mechanism that will reach the system equilibrium, or come as close as possible given the distributed decision-making structure. In general, however, we cannot expect to derive a system equilibrium or globally optimal solution without central control. Limits on coordination and communication may prevent the distributed resource allocation from exploiting all opportunities and inhibiting agents from acting at cross purposes. But under certain conditions decision making can indeed be decentralized effectively via market mechanisms. General-equilibrium analysis can help us to recognize and take advantage of these opportunities.

[3] The Nash equilibrium converges to the user equilibrium as the number of shippers increases and the effect of any individual's behavior on prices diminishes [6].

4 WALRAS: A Market-Oriented Programming Environment

To explore the use of market mechanisms for the coordination of distributed planning modules, I have developed a prototype environment for specifying and simulating computational markets. The system, called WALRAS,[4] provides basic mechanisms implementing various sorts of agents, auctions, and bidding protocols. To specify a computational economy, one defines a set of goods and instantiates a collection of agents that produce or consume those goods. The simulation engine of WALRAS then "runs" these agents to determine an equilibrium allocation of goods and activities in the economy.

4.1 Market Configuration and Equilibrium

Agents fall in two general classes. *Consumers* can buy, sell, and consume goods, and their preferences for consuming various combinations of goods are specified by their *utility function*. *Producers* can transform some sorts of goods into some others, according to their technology or *production function*. Each type of agent may start with an initial allocation of some goods, termed their *endowment*. The objective of a consumer is to maximize its utility, subject to the constraint (the *budget constraint*) that the cost of its consumption bundle does not exceed the value of its endowment at the going prices. The objective of a producer is to maximize profits given the going price of its output good and the inputs required for its production (the *factor goods*).

WALRAS associates an *auction* with each distinct good. Agents act in the market by submitting *bids* to auctions. The form of a bid is determined by the auction protocol. In a price mechanism, bids specify a correspondence between prices and quantities of the good that the agent offers to demand or supply. The auction derives a market-clearing price, at which the quantity demanded balances that supplied, within some prespecified tolerance. When the current price is clearing with respect to the current bids, we say the market for that commodity is in equilibrium.

[4] Named for the 19th-century French economist Léon Walras, who was the first to envision a system of interconnected markets in price equilibrium.

An agent acts competitively when it takes prices as given, neglecting any impact of its own behavior on the market-clearing price. Perfect competition reflects individual rationality when there are numerous agents, each small with respect to the entire economy. However, when an individual agent is large enough to affect prices significantly (i.e., possesses market power), it forfeits utility or profits by failing to take this into account.

Under the assumption of perfect competition, each agent's constrained optimization problem is parameterized by the prices of goods. We say that an agent is in equilibrium if its set of outstanding bids corresponds to the solution of its optimization problem at the going prices. If all the agents and commodity markets are in equilibrium, the allocation of goods dictated by the auction results is a *competitive equilibrium*. From the perspective of mechanism design, competitive equilibria possess several desirable properties, in particular, the two fundamental welfare theorems of general equilibrium theory: (1) all competitive equilibria are *Pareto optimal* (no agent can do better without some other doing worse), and (2) *any* Pareto optimum is a competitive equilibrium for some initial endowment. These properties seem to offer exactly what we need: a bound on the quality of the solution, plus the prospect that we can achieve the most desired behavior by carefully engineering the configuration of the computational market. Moreover, in equilibrium, the prices reflect exactly the information required for distributed agents to optimally evaluate perturbations in their behavior without resorting to communication or reconsideration of their full set of possibilities.

4.2 Computing Competitive Equilibria

Under certain "classical" assumptions (essentially continuity, monotonicity, and convexity of preferences and technologies; see, e.g., Varian [7]), competitive equilibria exist, and are unique given some further restrictions. They are also computable, and algorithms based on fixed-point methods [8] and optimization (variational inequality) techniques have been developed. Both sorts of algorithms in effect solve the simultaneous equilibrium equations by convergent iteration. However, by employing the equilibrium equations, these techniques violate the decentralization considerations underlying our distributed planning application.[5] For example, the constraint that profits be zero is a consequence of competitive behavior and constant-returns technology. Since information about the form of the technology and bidding policy is considered private to producer agents, it would not be permissible to embed the zero-profit condition into the equilibrium derivation procedure, as is sometimes done in computable general-equilibrium models. Similarly, explicitly examining the joint commodity space in the search for equilibrium undercuts our original motive for decomposing complex activities into consumption and production of separate goods.

WALRAS's procedure is a decentralized relaxation method, akin to the mechanism of *tatonnement* originally sketched by Léon Walras to explain how prices might be derived. A tatonnement method iteratively adjusts prices up or down as there is an excess of demand or supply, respectively (e.g., in proportion to the excess). The method employed by WALRAS successively computes an equilibrium price in each separate market, in a manner detailed below.[6] Like tatonnement, it involves an iterative adjustment of prices based on reactions of the agents in the market. However, it differs from traditional tatonnement procedures in that (1) agents submit supply and demand *curves* rather than single quantities for a particular price, and (2) the auction adjusts individual prices to clear, rather than the entire price vector by some function of summary statistics such as excess demand.

Figure 2 presents a schematic view of the WALRAS bidding process. There is an auction for each distinct good, and for each agent, a link to all auctions in which it has an interest. There is also a "tote board" of current prices, kept up-to-date by the various auctions.

Each agent maintains an agenda of bid tasks, specifying the markets in which it must update its bid or compute a new one. In Figure 2, agent A_j has pending tasks to submit bids to auctions G_1, G_7, and G_4. The bid for a particular good corresponds to one dimension of the agent's solution to its constrained optimization problem, which is parameterized by the prices for

[5] These methods are typically applied to the analysis of existing decentralized structures, such as transportation industries or even entire economies [9]. Because our purpose is to *implement* a distributed system, we must obey *computational* distributivity constraints not relevant to the usual purposes of applied general-equilibrium analysis.

[6] This general approach is called "progressive equilibration" by Dafermos and Nagurney [10]. WALRAS performs progressive equilibration in their sense, but adopts a different model of market structure in general, and transportation networks in particular.

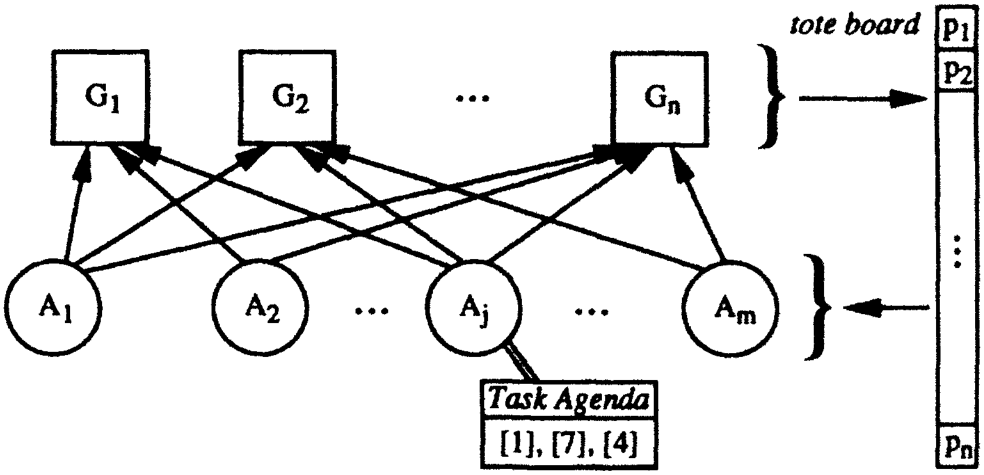

Figure 2: WALRAS's bidding process. G_i denotes the auction for the ith good, and A_i the ith trading agent.

all relevant goods. Acting as a perfect competitor, a WALRAS agent bids for a good under the assumption that prices for the remaining goods are fixed at their current values. The bid itself is a schedule of quantities and prices (encoded in any of a variety of formats) specifying the amount of the good demanded or supplied as its own price varies.

As new bids are received at auction, the previously computed clearing price becomes obsolete. Periodically, each auction computes a new clearing price (if any new or updated bids have been received) and posts it on the tote board. When a price is updated, this may invalidate some of an agent's outstanding bids, since these were computed under the assumption that prices for remaining goods were fixed. On finding out about a price change, an agent augments its task agenda to include the potentially affected bids.

At all times, WALRAS maintains a vector of going prices and quantities that would be exchanged at those prices. While the agents have nonempty bid agendas or the auctions new bids, some or all goods may be in disequilibrium. When all auctions clear and all agendas are exhausted, however, the economy is in competitive equilibrium (up to some numeric tolerance). This process is highly distributed, in that each agent need communicate directly only with the auctions for the goods of interest (those in the domain of its utility or production function, or for which it has nonzero endowments). Each of these interactions concerns only a single good; auctions never coordinate with each other. Agents need not negotiate directly with other agents, noı even know of each other's existence.

It is well known that tatonnement processes may not converge to equilibrium (but convergent results are indeed competitive equilibria) [8]. The class of economies in which tatonnement works are those with so-called *stable* equilibria: those without complementarities in preferences or technologies, where *gross substitutability* holds [11]. Using a recent result of Milgrom and Roberts [12, Theorem 12], it can be shown that WALRAS's adjustment process also converges under gross substitutability. The key observation is that in progressive equilibration (synchronous or not) the price at each time is based on some set of previous supply and demand bids.

Eydeland and Nagurney have analyzed in detail the convergence pattern of progressive equilibration algorithms similar to WALRAS for particular special cases [13].

4.3 Market-Oriented Programming

As described above, WALRAS provides facilities for specifying market configurations and computing their competitive equilibrium. We can also view WALRAS as a programming environment for decentralized resource allocation procedures. The environment provides constructs for specifying various sorts of agents and defining their interactions via their relations to common commodities. After setting up the initial configuration, the market can be run to determine the equilibrium level of activities and distribution of resources throughout the economy.

To cast a distributed planning problem as a market, one needs to identify (1) the goods traded, (2) the agents trading, and (3) the agents' bidding behavior. Finally, it might be advantageous to adjust some general parameters of the bidding protocol. These design steps are serially dependent, as the definition of what constitutes an exchangeable or producible commodity severely restricts the type of agents that it makes sense to include. Below, I illustrate the design task with a WALRAS formulation of the transportation example.

4.4 Implementation

WALRAS is implemented in Common Lisp and the Common Lisp Object System (CLOS). The current version provides basic infrastructure for running computational economies, including the underlying bidding protocol and a library of CLOS classes implementing a variety of agent types. The object-oriented implementation supports incremental development of market configurations. In particular, new types of agents can often be defined as slight variations on existing types, for example by modifying isolated features of the demand policy or bid format.

Although it models a distributed system, WALRAS runs serially on a single processor. Distribution constraints on information and communication are enforced by programming and specification conventions rather than by fundamental mechanisms of the software environment. Asynchrony is simulated by randomizing the bidding sequences so that agents are called on unpredictably. Indeed, artificial synchronization can lead to an undesirable oscillation in the clearing prices, as agents collectively overcompensate for imbalances in the preceding iteration.[7]

The current experimental system runs transportation models of the sort described in the next section, as well as some abstract exchange and production economies with parameterized utility and production functions (including the expository examples of Scarf [8] and Shoven and Whalley [9]). Customized tuning of the basic bidding protocol has not been necessary. In the process of getting WALRAS to run on these examples, I have produced some substantial intermediate object structure, but much more is required to fill out a comprehensive taxonomy of agents, bidding strategies, and auction policies.

[7] In some formal dynamic models [14, 15], homogeneous agents choose instantaneously optimal policies without accounting for others that are simultaneously making the same choice, detracting from its value. Oscillation occurs as the result of delayed feedback about the others' decisions. I have also observed this phenomenon empirically in a synchronized version of WALRAS. By eliminating the synchronization, agents tend to work on different markets at any one time, and hence do not suffer as much from this time delay.

5 WALRAS Transportation Market

In this section, I present a series of three transportation market structures implemented in WALRAS. The first and simplest model comprises the basic transportaticn goods and shipper agents, which are augmented in the succeeding models to include other agent types. Comparative analysis of the three market structures reveals the qualitatively distinct economic and computational behaviors realized by alternate WALRAS configurations.

5.1 Basic Shipper Model

The resource of primary interest in the multicommodity flow problem is movement of cargo. Because the value and cost of a cargo movement depends on location, we designate as a distinct good capacity on each origin-destination pair in the network (see Figure 1). To capture the cost or input required to move cargo, we define another good denoting generic transportation resources. In a more concrete model, these might consist of vehicles, fuel, labor, or other factors contributing to transportation.

To decentralize the decision making, we identify each movement requirement with a distinct shipper agent. These shippers, or consumers, have an interest in moving various units of cargo between specified origins and destinations.

The interconnectedness of agents and goods defines the market configuration. Figure 3 depicts the WALRAS configuration for the basic shipper model corresponding to the example network of Figure 1. In this model there are two shippers, $S_{1,4}$ and $S_{4,1}$, where $S_{i,j}$ denotes a shipper with a requirement to move goods from origin i to destination j. Shippers connect to goods that might serve their objectives: in this case, movement along links that belong to some simple path from the shipper's origin to its destination. In the diagram, $G_{i,j}$ denotes the good representing an amount of cargo moved over the link $i \rightarrow j$. G_0 denotes the special transportation resource good. Notice that the only goods of interest to both shippers are G_0, for which they both have endowments, and $G_{2,3}$, transportation on the link serving both origin-destination pairs.

The model we employ for transportation costs is based on a network with congestion, thus exhibiting diseconomies of scale. In other words, the marginal and average costs (in terms of transportation resources required)

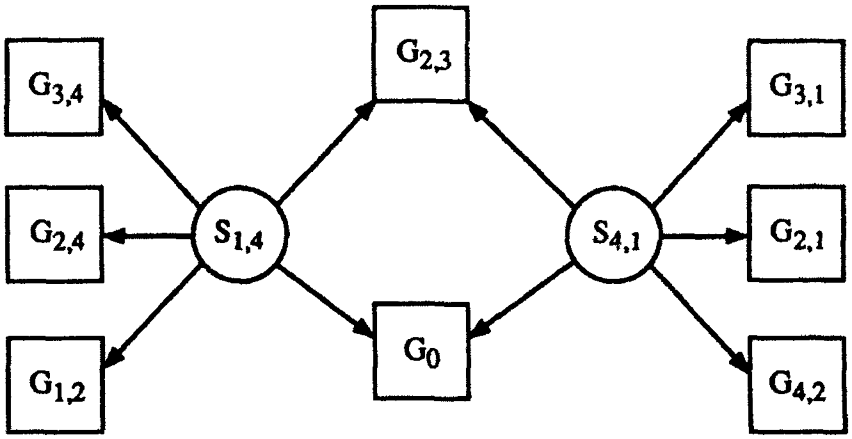

Figure 3: WALRAS basic shipper market configuration for the example transportation network.

are both increasing in the level of service on a link. Using Harker's data, costs are quadratic.[8] Let $c_{i,j}(x)$ denote the cost in transportation resources (good G_0) required to transport x units of cargo on the link from i to j. The complete cost functions are:

$$c_{1,2}(x) = c_{2,1}(x) = c_{2,4}(x) = c_{4,2}(x) = x^2 + 20x,$$

$$c_{3,1}(x) = c_{2,3}(x) = c_{3,4}(x) = 2x^2 + 5x.$$

Finally, each shipper's objective is to transport 10 units of cargo from its origin to its destination.

In the basic shipper model, we assume that the shippers pay proportionately (in units of G_0) for the total cost on each link. This amounts to a policy of average cost pricing.

We take the shipper's objective to be to ship as much as possible (up to its movement requirement) in the least costly manner. Given a network with prices on each link, the cheapest cargo movement corresponds to the

[8] The quadratic cost model is posed simply for concreteness, and does not represent any substantive claim about transportation networks. The important qualitative feature of this model is that it exhibits decreasing returns, a defining characteristic of congested networks. Note also that Harker's model is in terms of monetary costs, whereas we introduce an abstract input good.

shortest path in the graph, where distances are equated with prices. Thus, for a given link, a shipper would prefer to ship its entire quota on the link if it is on the shortest path, and zero otherwise. In the case of ties, it is indifferent among the possible allocations. To bid on link i, j, the shipper can derive the threshold price that determines whether it is on a shortest path by taking the difference in shortest-path distance between the networks where link i, j's distance is set to zero and infinity, respectively.

In incrementally changing its bids, the shipper should also consider its outstanding bids and the current prices. The value of reserving capacity on a particular link is zero if it cannot get service on the other links on the path. Similarly, if it is already committed to shipping cargo on a parallel path, it does not gain by obtaining more capacity (even at a lower price) until it withdraws these other bids.[9] Therefore, the actual demand policy of a shipper is to spend its uncommitted income on the potential flow increase (derived from maximum-flow calculations) it could obtain by purchasing capacity on the given link. It is willing to spend up to the threshold value of the link, as described above. This determines one point on its demand curve. If it has some unsatisfied requirement and uncommitted income it also indicates a willingness to pay a lower price for a smaller amount of capacity. Boundary points such as this serve to bootstrap the economy; from the initial conditions it is typically the case that no individual link contributes to overall flow between the shipper's origin and destination. Finally, the demand curve is completed by an arbitrary smoothing operation on these points.

WALRAS successfully computes the competitive equilibrium for this example, which in the case of the basic shipper model corresponds to a user equilibrium (UE) for the transportation network. In the UE for the example network, each shipper sends 2.86 units of cargo over the shared link 2 → 3, and the remaining cargo over the direct link from location 2 to the

[9] Even if a shipper could simultaneously update its bids in all markets, it would not be a good idea to do so here. A competitive shipper would send all its cargo on the least costly path, neglecting the possibility that this demand may increase the prices so that it is no longer cheapest. The outstanding bids provide some sensitivity to this effect, as they are functions of the own price. But they cannot respond to changes in many prices at once, and thus the policy of updating all bids simultaneously can lead to perpetual oscillation. For example, in the network considered here, the unique competitive equilibrium has each shipper routing portions of its cargo on all available paths. Policies allocating all cargo to one path can never lead to this result, and hence convergence to competitive equilibrium depends on the incrementality of bidding behavior.

destination. This allocation is inefficient, as its total cost is 1143 resource units, which is somewhat greater than the global minimum-cost solution of 1136 units. In economic terms, the cause of the inefficiency is an externality with respect to usage of the shared link. Because the shippers are effectively charged average cost—which in the case of decreasing returns is below marginal cost—the price they face does not reflect the full incremental social cost of additional usage of the resource. In effect, incremental usage of the resource by one agent is subsidized by the other. The steeper the decreasing returns, the more the agents have an incentive to overutilize the resource.[10]

The classical remedy to such problems is to internalize the externality by allocating ownership of the shared resource to some decision maker who has the proper incentives to use it efficiently. We can implement such a solution in WALRAS by augmenting the market structure with another type of agent.

5.2 Carrier Agents

We extend the basic shipper model by introducing *carriers*, agents of type producer who have the capability to transport cargo units over specified links, given varying amounts of transportation resources. In the model described here, we associate one carrier with each available link. The technology for each carrier is characterized by the cost function described above. To achieve a global movement of cargo, shippers obtain transportation services from carriers in exchange for the necessary transportation resources.

Let $C_{i,j}$ denote the carrier that transports cargo from location i to location j. Each carrier $C_{i,j}$ is connected to the auction for $G_{i,j}$, its output good, along with G_0—its input in the production process. Shipper agents are also connected to G_0, as they are endowed with transportation resources to exchange for transportation services. Figure 4 depicts the WALRAS market structure when carriers are included in the economy.

In the case of a decreasing returns technology, the producer's (carrier's) optimization problem has a unique solution. The optimal level of activity maximizes revenues minus costs, which occurs at the point where the output price equals marginal cost. Using this result, carriers submit supply bids

[10] Average-cost pricing is perhaps the most common mechanism for allocating costs of a shared resource. Shenker [16] points out problems with this scheme—with respect to both efficiency and strategic behavior—in the context of allocating access to congested computer networks, a problem analogous to our transportation task.

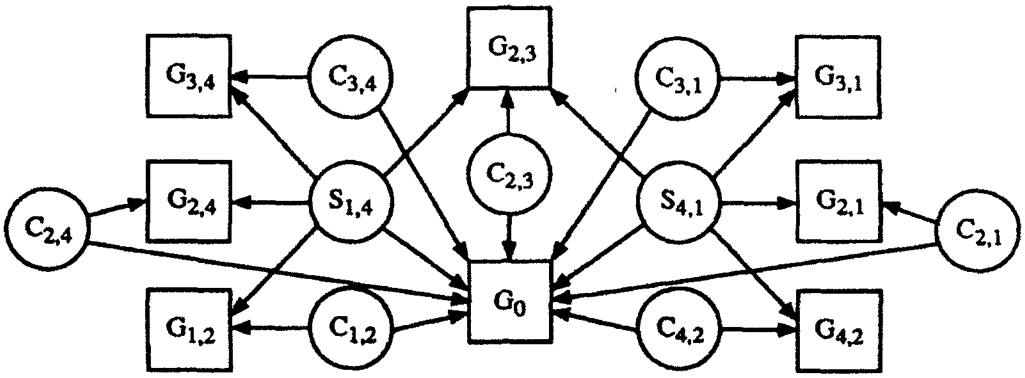

Figure 4: WALRAS market configuration for the example transportation network in an economy with shippers and carriers.

specifying transportation services as a function of link prices (with resource price fixed), and demand bids specifying required resources as a function of input prices (for activity level computed with output price fixed).

For example, consider carrier $C_{1,2}$. At output price $p_{1,2}$ and input price p_0, the carrier's profit is

$$p_{1,2}y - p_0c_{1,2}(y),$$

where y is the level of service it chooses to supply. Given the cost function above, this expression is maximized at $y = (p_{1,2} - 20p_0)/2p_0$. Taking p_0 as fixed, the carrier submits a supply bid with y a function of $p_{1,2}$. On the demand side, the carrier takes $p_{1,2}$ as fixed and submits a demand bid for enough good G_0 to produce y, where y is treated as a function of p_0.

With the revised configuration and agent behaviors described, WALRAS derives the system equilibrium (SE), that is, the cargo allocation minimizing overall transportation costs. The derived cargo movements are correct to within 10% in 36 bidding cycles, and to 1% in 72, where in each cycle every agent submits an average of one bid to one auction. The total cost (in units of G_0), its division between shippers' expenditures and carriers' profits, and the equilibrium prices[11] are presented in Table 1. Data for the UE

[11] As a simple check on these prices, we can verify that $p_{2,3} + p_{3,4} = p_{2,4}$ and $p_{2,3} + p_{3,1} = p_{2,1}$. Both these relationships must hold in equilibrium (assuming all links have nonzero movements), else a shipper could reduce its cost by rerouting some cargo. Indeed, for a simple (small and symmetric) example such as this, it is easy to derive the

solution of the basic shipper model are included for comparison. That the decentralized process produces a global optimum is perfectly consistent with competitive behavior—the carriers price their outputs at marginal cost, and the technologies are convex.

Table 1: Equilibria derived by WALRAS for the transportation example. TC, MC, and AC stand for total, marginal, and average cost, respectively. TC = shipper expense – carrier profit.

pricing	TC	exp.	prof.	$p_{1,2}$	$p_{2,1}$	$p_{2,3}$	$p_{2,4}$	$p_{3,1}$	$p_{3,4}$	$p_{4,2}$
MC (SE)	1136	1514	378	40.0	35.7	22.1	35.7	13.6	13.6	40.0
AC (UE)	1143	1143	0	30.0	27.1	16.3	27.1	10.7	10.7	30.0

The lesson from this exercise is that we can achieve qualitatively distinct results by simple variations in the market configuration or agent policies. From our designers' perspective, we prefer the configuration that leads to the more transportation-efficient SE. Examination of Table 1 reveals that we can achieve this result by allowing the carriers to earn nonzero profits (economically speaking, these are really rents on the fixed factor represented by the congested channel) and redistributing these profits to the shippers to cover their increased expenditures.[12]

5.3 Arbitrageur Agents

The preceding results demonstrate that WALRAS can indeed implement a decentralized solution to the multicommodity flow problem. But the market structure in Figure 4 is not as distributed as it might be, in that (1) all agents are connected to G_0, and (2) shippers need to know about all links potentially serving their origin-destination pair. The first of these concerns

equilibrium analytically using global equations such as these. But as argued above, it would be improper to exploit these relationships in the implementation of a truly distributed decision process.

[12] In the model of general equilibrium with production, consumers own shares in the producers' profits. This closes the loop so that all value is ultimately realized in consumption. We can specify these shares as part of the initial configuration, just like the endowment. In this example, we distribute the shares evenly between the two shippers.

is easily remedied, ⅽɜ the choice of a single transportation resource good was cumpletely arbitrary. For example, it would be straightforward to consider some collection of resources (e.g., fuel, labor, vehicles), and endow each shipper with only subsets of these.

The second concern can also be addressed within WALRAS. To do so, we introduce yet another sort of producer agent. These new agents, called ***arbitrageurs***, act as specialized middlemen, monitoring isolated pieces of the network for inefficiencies. An arbitrageur $A_{i,j,k}$ has two input goods, $G_{i,j}$ and $G_{j,k}$, and a single output, $G_{i,k}$. Its production function simply specifies that the amount of output is equal to the minimum of its two inputs. If $p_{i,j} + p_{j,k} < p_{i,k}$, then its production is profitable. Its bidding policy in WALRAS is to increment its level of activity at each iteration by an amount proportional to its current profitability (or decrement proportional to the loss). Such incremental behavior is necessary for all constant-returns producers in WALRAS, as the profit maximization problem has no interior solution in the linear case.

To incorporate arbitrageurs into the transportation market structure, we first create new goods corresponding to the transitive closure of the transportation network. In the example network, this leads to goods for every location pair. Next, we add an arbitrageur $A_{i,j,k}$ for every triple of locations such that (1) $i \rightarrow j$ is in the original network, and (2) there exists a path from j to k that does not traverse location i. The revised market structure for the running example is depicted in Figure 5, with new goods and agents shaded. Some goods and agents that are inactive in the market solution have been omitted from the diagram to avoid clutter.

Notice that in Figure 5 the connectivity of the shippers has been significantly decreased, as the shippers now need be aware of only the good directly serving their origin-destination pair. This dramatically simplifies their bidding problem, as they can avoid all analysis of the price network. The structure as a whole seems more distributed, as no agent is concerned with more than three goods.

Despite the simplified shipper behavior, WALRAS still converges to the SE, or optimal solution, in this configuration. Although the resulting allocation of resources is identical, a qualitative change in market structure here corresponds to a qualitative change in the degree of decentralization.

In fact, the behavior of WALRAS on the market configuration with arbitrageurs is virtually identical to a standard distributed algorithm for multi-

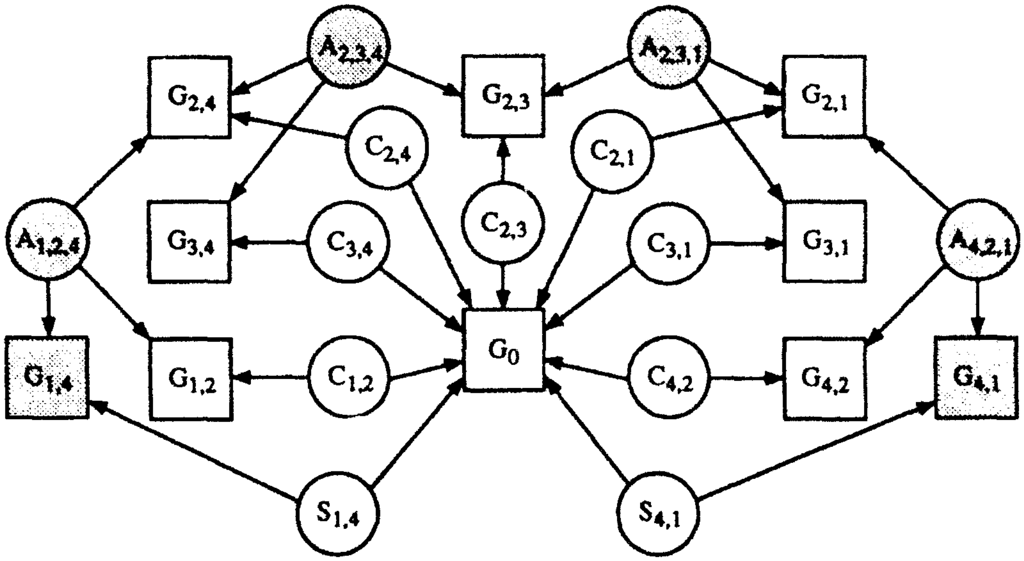

Figure 5: The revised WALRAS market configuration with arbitrageurs.

commodity flow (minimum delay on communication networks) due to Gallager [17]. Gallager's algorithm provably converges to the optimal solution as long as the iterative adjustment parameter is sufficiently small. Similarly, convergence in WALRAS for this model requires that the arbitrageurs do not adjust their activity levels too quickly in response to profit opportunities or loss situations.

6 (Some of the) Limitations

One serious limitation of WALRAS is the assumption that agents act competitively. There are two approaches toward alleviating this restriction in a computational economy. First, we could simply adopt models of imperfect competition, perhaps based on specific forms of imperfection (e.g., spatial monopolistic competition) or on general game-theoretic models. Second, as architects we can configure the markets to promote competitive behavior. For example, decreasing the agent's grain size and enabling free entry of agents should enhance the degree of competition. Perhaps most interestingly, by controlling the agents' knowledge of the market structure (via standard information-encapsulation techniques), we can degrade their ability to exploit whatever market power they possess. Uncertainty has been shown to increase competitiveness among risk-averse agents in some formal bidding models [18], and in a computational environment we have substantial control over this uncertainty.

The existence of competitive equilibria and efficient market allocations also depends critically on the assumption of nonincreasing returns to scale. Although congestion is a real factor in transportation networks, for many modes of transport there are often other economies of scale and density that may lead to returns that are increasing overall [19].

Having cast WALRAS as a general environment for distributed planning, it is natural to ask how universal "market-oriented programming" is as a computational paradigm. We can characterize the computational power of this model easily enough, by correspondence to the class of convex programming problems represented by economies satisfying the classical conditions. However, the more interesting issue is how well the conceptual framework of market equilibrium corresponds to the salient features of distributed planning problems. Although it is too early to make a definitive assertion about this, it seems clear that many planning tasks are fundamentally problems in resource allocation, and that the units of distribution often correspond well with units of agency. Economics has been the most prominent (and arguably the most successful) approach to modeling resource allocation with decentralized decision making, and it is reasonable to suppose that the concepts economists find useful in the social context will prove similarly useful in our analogous computational context. Of course, just as economics is not ideal for analyzing all aspects of social interaction, we should expect that many issues in the organization of distributed planning will not be well accounted-for in this framework.

Finally, the transportation network model presented here is a highly simplified version of the actual planning problem for this domain. A more realistic treatment would cover multiple commodity types, discrete movements, temporal extent, hierarchical network structure, and other critical features of the problem. Some of these may be captured by incremental extensions to the simple model, perhaps applying elaborations developed by the transportation science community.[13]

[13] For example, many transportation models (including Harker's more elaborate formulation [19]) allow for variable supply and demand of the commodities and more complex shipper-carrier relationships. Concepts of spatial price equilibrium, based on markets for commodities in each location, seem to offer the most direct approach toward extending the transportation model within WALRAS.

7 Related Work

The techniques and models described here obviously build on much work in economics, transportation science, and operations research. The intended research contribution here is not to these fields, but rather in their application to the construction of a computational framework for decentralized decision making in general and ultimately to distributed transportation planning in particular. Although the most elaborate WALRAS model is essentially equivalent to existing algorithms for distributed multicommodity flow, the market framework offers an approach toward extensions beyond the strict scope of this particular optimization problem.

The basic idea of applying economic mechanisms to coordinate distributed problem solving is not new to the AI community. Starting with the contract net [20], many have found the metaphor of markets appealing, and have built systems organized around markets or market-like mechanisms [21]. Miller and Drexler [22] have examined this approach in depth, presenting some underlying rationale and addressing specific issues salient in a computational environment. Waldspurger et al. [23] investigated the concepts further by actually implementing market mechanisms to allocate computational resources in a distributed operating system. For further remarks on this line of work, see [24].

Recently, Kuwabara and Ishida [25] have experimented with demand adjustment methods for a task very similar to the multicommodity flow problem considered here. Although they do not cast their system in a competitive-equilibrium framework, the results are congruent with those obtained by WALRAS.

WALRAS is distinct from these prior efforts in two primary respects. First, it is constructed expressly in terms of concepts from general equilibrium theory, to promote mathematical analysis of the system and facilitate the application of economic principles to architectural design. Second, WALRAS is designed to serve as a general programming environment for implementing computational economies. Although not developed specifically to allocate computational resources, there is no reason these could not be included in market structures configured for particular application domains. Indeed, the idea of grounding measures of the value of computation in real-world values (e.g., cargo movements) follows naturally from the general-equilibrium view of interconnected markets, and is one of the more exciting prospects for future applications of WALRAS to distributed problem-solving.

Organizational theorists have studied markets as mechanisms for coordinating activities and allocating resources within firms. For example, Malone [26] models information requirements, flexibility and other performance characteristics of a variety of market and non-market structures. In his terminology, WALRAS implements a *centralized market*, where the allocation of each good is mediated by an auction. Using such models, we can determine whether this gross form of organization is advantageous, given information about the cost of communication, the flexibility of individual modules, and other related features. In this paper, we examine in greater detail the coordination process in computational markets, elaborating on the criteria for designing decentralized allocation mechanisms. We take the distributivity constraint as exogenously imposed; when the constraint is relaxable, both organizational and economic analysis illuminate the tradeoffs underlying the mechanism design problem.

Finally, market-oriented programming shares with Shoham's [27] *agent-oriented programming* the view that distributed problem-solving modules are best designed and understood as rational agents. The two approaches support different agent operations (transactions versus speech acts), adopt different rationality criteria, and emphasize different agent descriptors, but are ultimately aimed at achieving the same goal of specifying complex behavior in terms of agent concepts (e.g., belief, desire, capability) and social organizations. Combining individual rationality with laws of social interaction provides perhaps the most natural approach to generalizing Newell's [28] "knowledge level analysis" idea to distributed computation.

8 Conclusion

In summary, WALRAS represents a general approach to the construction and analysis of distributed planning systems, based on general equilibrium theory and *competitive mechanisms*. The approach works by deriving the competitive equilibrium corresponding to a particular configuration of agents and commodities, specified using WALRAS's basic constructs for defining computational market structures. In a particular realization of this approach for a simplified form of distributed transportation planning, we see that qualitative differences in economic structure (e.g., cost-sharing among shippers versus ownership of shared resources by profit-maximizing carriers) correspond to qualitatively distinct behaviors (user versus system equilibrium). This exercise demonstrates that careful design of the distributed decision structure according to economic principles can sometimes lead to effective decentralization, and that the behaviors of alternative systems can be meaningfully analyzed in economic terms.

Acknowledgment

I have benefited from discussions of computational economies with several colleagues, and would like to thank in particular Jon Doyle, Eli Gafni, Anna Nagurney, Scott Shenker, Carl Waldspurger, and Martin Weitzman for helpful comments and suggestions.

9 References

[1] Leonid Hurwicz. The design of resource allocation mechanisms. In Arrow and Hurwicz [11], pages 3-37. Reprinted from *American Economic Review Papers and Proceedings*, 1973.

[2] Stanley Reiter. Information incentive and performance in the (new)^2 welfare economics. In Stanley Reiter, editor, *Studies in Mathematical Economics*. MAA Studies in Mathematics, 1986.

[3] W. Hildenbrand and A. P. Kirman. *Introduction to Equilibrium Analysis: Variations on Themes by Edgeworth and Walras*. North-Holland Publishing Company, Amsterdam, 1976.

[4] Patrick T. Harker. Multiple equilibrium behaviors on networks. *Transportation Science*, 22:39-46, 1988.

[5] Patrick T. Harker. Alternative models of spatial competition. *Operations Research*, 34:410-425, 1986.

[6] A. Haurie and P. Marcotte. On the relationship between Nash-Cournot and Wardrop equilibria. *Networks*, 15:295-308, 1985.

[7] Hal R. Varian. *Microeconomic Analysis*. W. W. Norton & Company, New York, second edition, 1984.

[8] Herbert E. Scarf. The computation of equilibrium prices. In Herbert E. Scarf and John B. Shoven, editors, *Applied General Equilibrium Analysis*, pages 1-49. Cambridge University Press, Cambridge, 1984.

[9] John B. Shoven and John Whalley. Applied general-equilibrium models of taxation and international trade: An introduction and survey. *Journal of Economic Literature*, 22:1007-1051, 1984.

[10] Stella Dafermos and Anna Nagurney. Supply and demand equilibration algorithms for a class of market equilibrium problems. *Transportation Science*, 23:118-124, 1989.

[11] Kenneth J. Arrow and Leonid Hurwicz, editors. *Studies in Resource Allocation Processes*. Cambridge University Press, Cambridge, 1977.

[12] Paul Milgrom and John Roberts. Adaptive and sophisticated learning in normal form games. *Games and Economic Behavior*, 3:82–100, 1991.

[13] Alexander Eydeland and Anna Nagurney. Progressive equilibration algorithms: The case of linear transaction costs. *Computer Science in Economics and Management*, 2:197–219, 1989.

[14] B. A. Huberman, editor. *The Ecology of Computation*. North-Holland, 1988.

[15] J. O. Kephart, T. Hogg, and B. A. Huberman. Dynamics of computational ecosystems. *Physical Review A*, 40:404–421, 1989.

[16] Scott Shenker. Congestion control in computer networks: An exercise in cost-sharing. Prepared for delivery at Annual Meeting of the American Political Science Association, August 1991.

[17] Robert G. Gallager. A minimum delay routing algorithm using distributed computation. *IEEE Transactions on Communications*, 25:73–85, 1977.

[18] R. Preston McAfee and John McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, 1987.

[19] Patrick T. Harker. *Predicting Intercity Freight Flows*. VNU Science Press, Utrecht, The Netherlands, 1987.

[20] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.

[21] T. W. Malone, R. E. Fikes, K. R. Grant, and M. T. Howard. Enterprise: A market-like task scheduler for distributed computing environments. In Huberman [14], pages 177–205.

[22] Mark S. Miller and K. Eric Drexler. Markets and computation: Agoric open systems. In Huberman [14], pages 133–176.

[23] Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffrey O. Kephart, and Scott Stornetta. Spawn: A distributed computational economy. *IEEE Transactions on Software Engineering*, 18:103–117, 1992.

[24] Michael P. Wellman. Review of Huberman [14]. *Artificial Intelligence*, 52:205–218, 1991.

[25] Kazuhiro Kuwabara and Toru Ishida. Symbiotic approach to distributed resource allocation: Toward coordinated balancing. In *Pre-Proceedings of the 4th European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, July 1992.

[26] Thomas W. Malone. Modeling coordination in organizations and markets. *Management Science*, 33:1317–1332, 1987.

[27] Yoav Shoham. Agent-oriented programming. Technical Report STAN-CS-90-1335, Stanford University Computer Science Department, 1990.

[28] Allen Newell. The knowledge level. *Artificial Intelligence*, 18:87–127, 1982.

U.S. GOVERNMENT PRINTING OFFICE 750-113